一.用户协同过滤

1.首先得到用户对每个Item的评分，由于item很多。这可能是个稀疏矩阵，因为用户不可能访问每个item

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Item1 | Item2 | Item3 | Item4 | Item5 | Item6 |
| 用户1 | 3.5 |  |  | 5 |  |  |
| 用户2 |  | 2 |  |  | 6 |  |

2.每个用户筛选出数据作为一个向量

用户1[3.5,0,0,5,0,0]

用户2[0,2,0,0,6,0]

计算出这2个向量的余弦相似度在(0,1)之间，然后得到所有玩家之间的相似度

2个向量的余弦相似度计算公式=v1点乘v2/(|v1|\*|v2|)

向量求模的计算公式：

Vec={v1,v2,v3} = sqrt(v1\*v1+v2\*v2+v3\*v3)

|  |  |  |
| --- | --- | --- |
|  |  | 相似度 |
| 用户1 | 用户2 | 0.8 |
| 用户2 | 用户3 | 0.2 |

3.筛选出和目标用户相似度最高的n个用户

4.根据这些预选的用户来对每个物品计算评分；公式是

(用户相似度(n1,target)\*用户1物品评分+用户相似度(n2,target)\*用户2物品评分…….)/( 用户1物品评分+用户2物品评分)

计算出每个物品的预测评分

最后对物品评分进行排序。得分最高的进行推荐

二．基于ALS的协同过滤召回

在下图中是一个稀疏矩阵，ALS算法的目标是通过算法把这个稀疏矩阵填充成一个完整的矩阵

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Item1 | Item2 | Item3 | Item4 | Item5 | Item6 |
| 用户1 | 3.5 |  |  | 5 |  |  |
| 用户2 |  | 2 |  |  | 6 |  |

假设这个据称是m\*n列，那么要构造2个中间矩阵，维度分别是m\*k和k\*n，这样这2个矩阵的叉乘结果是m\*n,这2个矩阵里面都是随机的参数

叉乘后的矩阵和正式的矩阵做最小差值回归，求得最佳参数，

有了最佳参数，就相当于得到了一个预测最佳的完整矩阵

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Item1 | Item2 | Item3 | Item4 | Item5 | Item6 |
| 用户1 | 3.5 | 3 | 4 | 5 | 4 | 3.5 |
| 用户2 | 2 | 2 | 1.5 | 5 | 6 | 4 |

预测出来的矩阵中每一个值就是用户的推荐度